

RDOS

Instruction Manual

five dollars

RDOOS

© Copyright 1978. All Rights Reserved.

 **Cromemco**
i n c o r p o r a t e d
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043

Table of Contents

	<i>Pg. No.</i>
Introduction	1
Command Format	
Swath Operator	
Errors and Escapes	
Baud Rate Selection	
System Stack	
Using the RDOS Monitor	
Commands	2
Boot	
Display Memory	
Examine Input Port	
Go	
Initialize Baud Rate	
Kick Stack	
Move	
Output	
Read Disk	
Seek Track	
Substitute Memory	
Verify	
Write Disk	
Select Disk Drive	
An Illustrative Example.	6
RDOS Program Listing	7

Introduction

The Cromemco Resident Disk Operating System (RDOS) is a 1K-byte program supplied in ROM with each Cromemco model 4FDC disk controller card. The RDOS program is designed to execute beginning at location C000 in memory space.

RDOS includes a bootstrap loader for the Cromemco Disk Operating System (CDOS) supplied with Cromemco disk software. RDOS also includes a system monitor with over a dozen commands. Several of the RDOS monitor commands are the same as those used in the Cromemco Z-80 Monitor program. These include commands to transfer program control, display the contents of memory, change the contents of memory, move and compare blocks of memory, write data to output ports, and change the baud rate of the serial port on the 4FDC card. The RDOS monitor also has several unique commands designed specifically for disk operations. These include commands to select one of four disk drives, set the head seek rate, seek disk tracks, read blocks of data from the disk, and write blocks of data onto the disk.

To use RDOS, first be sure that the power-on jump address of your ZPU card is set to memory location C000 (see Table 1). If you wish to boot in CDOS from your disk without entering the RDOS monitor, then set Switch 3 of the 4FDC card to the ON position (this selects auto-boot mode). Once CDOS is booted in, you may return to the RDOS monitor by executing the "BYE" command in CDOS. If on power-up or reset you do wish to enter the RDOS monitor, then set Switch 3 of the 4FDC card to the OFF position (see Table 2). Once the RDOS monitor is entered, you can begin using the RDOS commands described in this manual.

ZPU Switch	Switch Position
A15	1
A14	1
A13	0
A12	0

Table 1

The power-on jump address switch on the ZPU card should be set as shown to begin automatic execution of RDOS at location C000 in memory space.

RDOS Mode	4FDC Switch 3
CDOS Bootstrap	ON
RDOS Monitor	OFF

Table 2

The setting of Switch 3 on the 4FDC card sets the RDOS mode of operation on power up. When this switch is ON, CDOS is automatically booted in from the system diskette. When this switch is OFF, the RDOS Monitor is entered.

Command Format

The normal prompt of the monitor is a semi-colon, ';'. However, if a disk drive is selected the prompt changes in order to remind the user which drive is current. (See Select Disk Drive for details.)

The monitor is controlled by one and two-character commands from the terminal keyboard. The format is free-form with respect to spaces. All data is entered and printed in hexadecimal format.

In the following, DM is the Display Memory command and S is the Swath operator (see below). The four examples are equivalent commands. They display the contents of 100 hex bytes of memory beginning with location 1000 hex. ('(CR)' indicates carriage return).

```
;DM1000 10FF (CR)
;DM1000S100 (CR)
; D M 1000 10FF (CR)
; D M 1000 S 100 (CR)
```

When entering an address as an operand, only the last four digits typed in are retained. For example, '321000' is read as '1000'. Therefore, if a wrong digit is entered, continue typing until the last four digits are correct.

Only the last two digits typed are retained when a two-digit number such as a data byte is entered.

Swath Operator

There are two ways to specify the address range of many commands. The first is to simply list the beginning and ending addresses (and, where appropriate, the destination address). For example, the first command displays the contents of memory between addresses E400 and E402. The second com-

mand moves (or copies) the first 1400 hex bytes of memory to memory starting at 2000 hex.

```
DM E400 E402
M 0 13FF 2000
```

Another way to do the same thing is to use the Swath operator, S, to specify the width of the address range rather than state the ending address explicitly.

```
DM E400S3
M 0 S1400 2000
```

Errors and Escapes

When the monitor detects an error condition, the command is aborted and a '?' is printed followed by the prompt ';' for the next command.

Any command may be aborted from the keyboard either when the monitor is requesting further input, or during print-out, by depressing either of the ESCAPE or the ALT MODE key. (CONTROL-SEMI-COLON, CONTROL-SHIFT-'K', and '}' may also work, depending on the design of your CRT terminal.)

Baud Rate Selection

When the monitor is entered, push carriage-return (up to four times) until the monitor responds with:

```
CROMEMCO RDOS
```

The monitor is capable of selecting 19200, 9600, 4800, 2400, 1200, 300, 150, or 110 baud. The maximum number of carriage-returns required to select any of these baud rates is four.

The baud rate can also be changed by using Initialize command. (See page 3.)

Some peripheral devices such as paper tape readers or punches may have no keyboards. The baud rate can also be set by outputting a data byte from the following table to port 0.

Baud Rates	Data Byte
110	01
150	82
300	84
1200	88
2400	90
4800	A0
9600	C0

The baud rate can be octupled by outputting 10 hex to port 2. Outputting 0 to this port brings the baud rate back to normal.

System Stack

The RDOS stack normally resides in low memory between 40 and 80 hex. However, if it is in the way, it can be moved using the Kick Stack command. (See page 3.)

Using the RDOS Monitor

Set the power-on jump switch on the ZPU card to C (1100 binary) and turn off DIP switch 3 on the 4FDC.

Depress carriage-return two to four times in order to set the UART on the 4FDC to the baud rate of the terminal being used.

The monitor will then respond:

```
CROMEMCO RDOS
```

followed by a prompt ';'. The monitor is then ready to accept commands from the keyboard.

Commands

Boot

- (1) B (CR)

Boots CDOS from the diskette on drive-A. CDOS will then respond with its prompt 'A.'.

Display Memory

- (2a) DM beginning-addr ending-addr (CR)
or
- (2b) DM beginning-addr S swath-width

The contents of memory are displayed in hexadecimal form. Each line of the display is preceded by the address of its first byte. For example:

```
;DM100 S3
0100 : C3 34 7F
```

Examine Input Port

- (3) E port-number (CR)

Displays the current contents of the input port identified by port-number (in hex).

Go

- (4) G starting-addr (CR)

Execution begins at starting-address.

Initialize Baud Rate

(5) I (CR)

After the carriage-return is typed, change the baud rate of the terminal to the desired value and then push carriage-return until the monitor responds with its prompt.

The monitor is capable of selecting 19200, 9600, 4800, 2400, 1200, 300, 150, or 110 baud. The maximum number of carriage returns required to select any of these baud rates is four.

Kick Stack

(6) K new-stack-location (CR)

Moves the monitor's stack from normal location at 7C hex to any convenient location in RAM memory. Remember to leave 64 (40 hex) bytes for the system stack above its new location (including 4 bytes for temporary storage above the stack proper).

Move

(7a) M source-addr source-end destination-addr (CR)
or

(7b) M source-addr S swath-width destination-addr (CR)

Move (or copy) the contents of memory beginning with source-address and ending with source-end to destination-address. After the move, the monitor verifies that source and destination are the same. This will result in a print-out of discrepancies which are not really errors after certain types of overlapping moves. However, this print-out can be terminated by depressing ESCAPE or ALT Mode.

The move command can be used to fill a block of memory with a constant. For example, to enter zeros between locations 100 and 108, use the Substitute Memory command to enter 0 at location 100, and then move 100 through 107 to 101:

M 100 107 101

or

M 100 S 8 101

Care should be taken not to overwrite the monitor's stack which resides in low memory between 40 and 80 hex unless changed with the Kick Stack command.

Output

(8) O data-byte port-number (CR)

Writes data to the output port identified by port-number (in hex).

Read Disk

(9a) RD destination-addr destination-end sector-number (CR)

or

(9b) RD destination-addr S swath-width sector-number (CR)

Before this command will be accepted the disk drive and track number must have been specified. (See the Select Disk Drive and Seek commands.)

This command reads enough sectors from the current drive to fill the specified memory area, starting with the specified sector of the current track. The first track and sector and the last track and sector read are then displayed. However, if the last sector of the last track on the diskette is read before the memory area is filled then a question mark is printed and the command is terminated. The track and sector designations for both 5" and 8" diskettes are shown in Table 3.

The command is also terminated if an error occurs in reading a sector. In this case, a message of the following type is printed:

R-ERR nn

where nn is a hex number which indicates the status:

Bit	Indication
7	Not Ready
6	Record Type
5	Record Type
4	Record Not Found
3	CRC Error
2	Lost Data
1	Data Request
0	Busy

The number of the last track accessed can be obtained from input port 31 hex and the number of the last sector accessed from input port 32 hex. (See the Examine Input Port command.)

	8" Diskette	5" Diskette
Tracks	0-4C hex	0-27 hex
Sectors	1-1A hex	1-12 hex

Table 3

Care should be taken not to overwrite the monitor's stack which normally resides in lower RAM

between 40 and 80 hex. If it is desired to load this region of memory from the disk, first move the stack using the Kick Stack command.

Seek Track

(10) S track-number (CR)

Before this command will be accepted the disk drive must be specified. (See the Select Disk Drive command.)

This command seeks the specified track of the current drive.

If an error is made, a message of the following type is printed:

S-ERR nn

where nn is a hex number which indicates the status:

Bit	Indication
7	Not Ready
6	Write Protect
5	Head Engaged
4	Seek Error
3	CRC Error
2	Track 0
1	Index
0	Busy

Substitute Memory

(11) SM address (CR)

Substitute Memory displays the contents of address and outputs a dot, '.', as a prompt for the substituted value. If no change is desired, type a space or another dot. Otherwise, enter the new value. The monitor accepts hex digits until it gets a delimiter, such as a space, dot, or carriage-return, retaining the last two digits entered as the value. Unless the delimiter is a carriage-return, the monitor then outputs the contents of the next sequential memory location with a dot prompt. A carriage-return terminates the command.

Verify

(12a) V source-addr source-end destination-addr (CR)
or

(12b) Vsource-addr S swath-width destination-addr(CR)

This command verifies that the block of memory between source-address and source-end contains the

same values as the block beginning at destination-address. The addresses and contents are printed for each discrepancy found (unless the print-out is terminated by ESCAPE or ALT MODE).

This command works by reading bytes from the source and destination and comparing them. If a discrepancy is found, the memory is read again for print-out. Thus it can happen that a discrepancy is printed-out with the source and destination contents indicated to be the same. This is caused by a defective memory element.

Write Disk

(13a) WD source-addr source-end sector-number (CR)
or

(13b) WD source S swath-width sector-number (CR)

Before this command will be accepted the disk drive and track number must have been specified. (See the Select Disk Drive and Seek commands).

This command writes the contents of the specified memory area to the current drive, starting with the specified sector of the current track. The first track and sector and the last track and sector written are then displayed (see Table 3). However, if part of the memory area remains after the last sector of the last track is written, a question mark is printed and the command is terminated.

The command is also terminated if an error is made in writing a sector. In this case, a message of the following type is made:

W-ERR nn

where nn is a hex number indicating the status:

Bit	Indication
7	Not Ready
6	Write Protect
5	Write Fault
4	Record Not Found
3	CRC Error
2	Lost Data
1	Data Request
0	Busy

The last track accessed can be obtained from port 31 hex. The last sector accessed from port 32 hex. (See Examine Input command.)

Select Disk Drive

The 4FDC will control up to four disk drives, labelled "A", "B", "C", and "D". It can handle seek speeds from the slow seek appropriate to the mini floppy to the fast seek of Cromemco's large floppy. It can also handle the medium seek of some other large floppies. To select a drive and a seek speed, type the drive label followed by one semi-colon for the fast seek, and two semi-colons for medium seek, or three semi-colons for slow seek. For example, to select drive C with slow seek, type:

```
C ; ; ; (CR)
```

To select drive A with fast seek, type:

```
A ; (CR)
```

Until the drive selection is changed the normal monitor prompt, ';;', will be replaced by the disk label and speed indicator as typed, 'C ; ; ;' in the first example.

All disk commands (Seek, Read Disk, and Write Disk) refer to the drive most recently selected.

Disk selection also restores the disk drive head to home, track 0. If an error is made in doing this a message of the following type is printed:

```
H-ERR nn
```

where nn is a hex number indicating the status:

Bit	Indication
7	Not Ready
6	Write Protect
5	Head Engaged
4	Seek Error
3	CRC Error
2	Track 0
1	Index
0	Busy

An Illustrative Example

To illustrate a specific use of the RDOS commands, consider the situation where you wish to make a copy of a diskette. If you have two disk drives this can easily be done using the CDOS command XFER. But if you have only one disk drive, you can use RDOS to read the original diskette into memory, and then write from memory to the new diskette. Since the total amount of system RAM memory is typically less than the capacity of a diskette, the procedure will have to be repeated several times — a different portion of the diskette is copied with each iteration.

The following procedure explains how to use RDOS to make a back-up mini-diskette using only one drive. This procedure assumes that there is 32K of contiguous low memory; if the user has less memory, the procedure

can easily be adapted to a smaller configuration by decreasing the swath lengths of the READS and WRITES. Note that you must also keep track of the sector numbers when changing the following procedure. This is easily done by noting the starting and stopping track and sector numbers given after the completion of a READ or WRITE.

Throughout the following, note that the commands which are typed by the user are underlined. The other prompts and messages are those issued by RDOS. The messages enclosed in brackets are parenthetical remarks and should not be typed into the terminal.

Get into RDOS and insert the disk to be copied, or the original disk, into the drive. Then type the following RDOS commands.

```
A:;;  
A:;;S0  
A:;;RD100 SZE00 1  
0001 0D12
```

[INSERT BLANK DISK WHICH HAS BEEN INITIALIZED AND WHICH WILL BE THE NEW BACK-UP DISK.]

```
A:;;S0  
A:;;WD100 SZE00 1  
0001 0D12
```

[INSERT ORIGINAL DISK.]

```
A:;;SE  
A:;;RD100 SZE00 1  
0E01 1B12
```

[INSERT BACK-UP DISK.]

```
A:;;SE  
A:;;WD100 SZE00 1  
0E01 1B12
```

[INSERT ORIGINAL DISK.]

```
A:;;S1C  
A:;;RD100 S6C00 1  
1C01 2712
```

[INSERT BACK-UP DISK.]

```
A:;;S1C  
A:;;WD100 S6C00 1  
1C01 2712
```

```
A:;;S0  
A:;;E
```

[HOME THE HEAD.]
[SYSTEM SHOULD BOOT UP INTO CDOS FROM THE NEW BACK-UP DISK. TYPE THE DIRECTORY AND VERIFY THAT THE TRANSFER IS CORRECT AND COMPLETE BY TRYING SEVERAL OF THE FILES.]

RDOS Program Listing

```

0000      0002 ; COPYRIGHT (C) 1977, CROMEMCO, INC.
          0003 ;
          0004 ;
          0005      ORG      0C000H ;START OF PROM
          0006 ;
          0007 ;
(007C)    0008 STACK: EQU      7CH      ;MUST LEAVE ROOM FOR
          0009 ;                      ;4 BYTES OF TEMP STORAGE
          0010 ;                      ;ABOVE THE STACK
          0011 ;                      ;(STACK) = DISK FLAGS
          0012 ;                      ;(STACK+1) = DISK LETTER (A - D)
          0013 ;                      ;(STACK+2) - (STACK+3): ROOM FOR
          0014 ;                      ;UP TO 2 SEMI-COLONS AS PART OF
          0015 ;                      ;THE DISK PROMPT.
          0016 ;
          0017 ;
(0004)    0018 NDRIVES: EQU      4          ;MAX. NO. OF DISK DRIVES
          0019 ;
          0020 ; BIT ASSIGNMENT FOR THE DISK FLAGS
          0021 ;
(0007)    0022 FASTSEEK: EQU      7
(0005)    0023 DISKMODE: EQU      5
(0004)    0024 MAXI: EQU      4
          0025 ;THE DISK NUMBER (0 - 3) OCCUPIES BITS 0 & 1
          0026 ;
          0027 ;
(0030)    0028 DSTAT: EQU      30H      ;DISK STATUS PORT
(0030)    0029 DCOMMND: EQU      30H      ;DISK COMMAND PORT
(0032)    0030 DSEC: EQU      32H      ;DISK SECTOR PORT
(0033)    0031 DDATA: EQU      33H      ;DISK DATA PORT
(0034)    0032 DFLAGS: EQU      34H      ;DISK FLAGS PORT
(0034)    0033 DCONTR: EQU      34H      ;DISK CONTROL PORT
(0031)    0034 DTRACK: EQU      31H      ;DISK TRACK PORT
          0035 ;
(0003)    0036 IMASK: EQU      3          ;INTERRUPT MASK PORT
(0000)    0037 BAUD: EQU      0          ;BAUD RATE PORT
(0004)    0038 PARLEL: EQU      4          ;PARALLEL PORT
(0040)    0039 BOOTSW: EQU      40H      ;BOOT SWITCH
(0010)    0040 MAXIM: EQU      10H      ;MASK FOR MAXI DISK
(0020)    0041 HDLDM: EQU      20H      ;HEAD LOAD MASK
          0042 ;
          0043 ;
(0000)    0044 STAT: EQU      0          ;STATUS PORT
(0001)    0045 DATA: EQU      1          ;DATA PORT
(0002)    0046 COMMND: EQU      2          ;COMMAND PORT
(0040)    0047 DAV: EQU      40H      ;DATA-AVAILABLE MASK
(0080)    0048 TBE: EQU      80H      ;XMITTER-BUF-EMPTY MSK
          0049 ;
(0000)    0050 CASE: EQU      0
          0051 ;
(000D)    0052 CR: EQU      0DH
(000A)    0053 LF: EQU      0AH
(001B)    0054 ESC: EQU      1BH
(007D)    0055 ALT: EQU      7DH
          0056 ;

```

```

0057 ;
0058 ;+++++
0059 ;
C000 217C00 0060 START: LD HL,STACK
C003 F9 0061 LD SP,HL
C004 EB 0062 EX DE,HL ;DE -> TEMP STORAGE
C005 F3 0063 DI
C006 CDEEC0 0064 CALL INITBAUD ;INIT. THE SERIAL PORT
C009 97 0065 SUB A
C00A D303 0066 OUT IMASK,A ;MASK OUT 4FDC INTERRUPTS
C00C DB34 0067 IN A,DFLAGS ;READ DISK FLAGS
C00E E640 0068 AND BOOTSW ;LOOK AT BOOT SWITCH
C010 2806 0069 JR Z,BOOTDK
C012 C30EC1 0070 JF MONITR
0071 ;
0072 ;
0073 ; MONITOR COMMAND
0074 ; QUIT THE MONITOR & BOOT CDOS IN.
0075 ;
C015 0076 BOOTMC:
C015 CD29C2 0077 CALL SKSGCR ;REQUIRE A CR
0078 ;
0079 ;
0080 ; BOOT DISK
0081 ;
C018 0082 BOOTDK:
C018 3ED0 0083 LD A,0D0H ;TERMINATE THE HOMING
C01A D330 0084 OUT DCOMMND,A ;OF THE DISK HEAD
C01C DB30 0085 BOT200: IN A,DSTAT
C01E 1F 0086 RRA
C01F 38FB 0087 JR C,BOT200
C021 F3 0088 DI
C022 3E10 0089 LD A,1 SHL MAXI ;MAXI FLAG
C024 0090 BOT300:
C024 218000 0091 LD HL,0080H ;INIT. BUFFER PNTR
C027 F9 0092 LD SP,HL ;& STACK PNTR
C028 F5 0093 PUSH AF ;SAVE MINI/MAXI FLAG
C029 44 0094 LD B,H ;0 [DISK A]
C02A CD3FC0 0095 CALL DHOME ;HOME DISK
C02D 200B 0096 JR NZ,BOT500 ;DISK ERROR
C02F F1 0097 POP AF ;GET MINI/MAXI FLAG
C030 F5 0098 PUSH AF
C031 44 0099 LD B,H ;0 [DISK A]
C032 1E01 0100 LD E,1 ;SECTOR 1
C034 CD8CC0 0101 CALL DREAD ;READ THE SECTOR
C037 CA8000 0102 JF Z,80H ;OK, GO EXECUTE
C03A F1 0103 BOT500: POP AF ;GET MINI/MAXI FLAG
C03B EE10 0104 XOR 1 SHL MAXI ;TOGGLE IT
C03D 18E5 0105 JR BOT300
0106 ;
0107 ;
0108 ;HOME DISK DRIVE
0109 ;
0110 ;INPUT - B CONTAINS DISK NUMBER (0,1,2,3)
0111 ; A BIT 4 CONTAINS 1 IF MAXI
0112 ;
0113 ;OUTPUT - B CONTAINS STATUS
0114 ; ZERO FLAG RESET IF ERROR
0115 ;
0116 ;REGISTERS A,F,B,C ARE CHANGED
0117 ;
0118 ;

```

```

C03F CDCBC0      0119 DHOME:  CALL    SELECT      ;SELECT DISK
C042 D334        0120          OUT    DCONTR,A      ;OUTPUT SELECT BYTE
C044 1698        0121          LD     D,98H         ;ERROR MASK
C046 E610        0122          AND    MAXIM         ;MAXI DISK?
C048 3E7F        0123          LD     A,7FH         ;TURN OFF HIGH SPEED SEEK
C04A D304        0124          OUT    PARLEL,A      ;
C04C 3E0F        0125          LD     A,0FH         ;LOAD MINI RESTORE COMMAND
C04E 281A        0126          JR     Z,EXECUTE     ;NO, ITS A MINI
C050 3E0D        0127          LD     A,0DH         ;MAXI RESTORE COMMAND
C052 1816        0128          JR     EXECUTE       ;EXEC COMMAND &
                                0129 ;             ;WAIT TIL DONE
                                0130 ;
                                0131 ;
                                0132 ;SEEK TO DESIRED TRACK
                                0133 ;
                                0134 ;TRACK REGISTER MUST HAVE BEEN PREVIOUSLY LOADED
                                0135 ;(MAY BE DONE BY INITIALLY DOING A HOME)
                                0136 ;
                                0137 ;INPUT -- B CONTAINS DISK DRIVE (0,1,2,3)
                                0138 ;           D CONTAINS TRACK
                                0139 ;           A BIT 7 = 1 FOR FAST SEEK
                                0140 ;           A BIT 4 = 1 FOR MAXI
                                0141 ;
                                0142 ;OUTPUT -- B CONTAINS STATUS
                                0143 ;           ZERO FLAG RESET IF ERROR
                                0144 ;
                                0145 ;REGISTERS A,F,B,C,D ARE CHANGED
                                0146 ;
C054 F5         0147 DSEEK:  PUSH    AF             ;SAVE DISK FLAGS
C055 CDCBC0     0148          CALL    SELECT      ;SELECT DISK
C058 D334       0149          OUT    DCONTR,A      ;OUTPUT CONTROL BYTE
C05A ED51       0150          OUT    (C),D         ;OUTPUT DESIRED TRACK
C05C 1698       0151          LD     D,98H         ;ERROR MASK
C05E F1         0152          POP     AF           ;GET FLAGS
C05F 17         0153          RLA                ;FAST SEEK?
C060 3814       0154          JR     C,DSK500      ;
C062 E620       0155          AND    MAXIM SHL 1    ;MASK FOR MINI/MAXI
C064 3E1F       0156          LD     A,01FH        ;LOAD SEEK COMMAND FOR MINI
C066 2802       0157          JR     Z,EXECUTE     ;MINI DISK
C068 3E1D       0158          LD     A,1DH         ;LOAD COMMAND FOR MAXI
                                0159 ;
                                0160 ;
C06A           0161 EXECUTE:
C06A D330       0162          OUT    DCOMMND,A      ;OUTPUT COMMAND
                                0163 ;
C06C           0164 EXCCHK:
C06C DB34       0165          IN     A,DFLAGS       ;WAIT FOR COMPLETION
C06E 1F         0166          RRA                ;
C06F 30FB       0167          JR     NC,EXCCHK      ;UNTIL INTREQ
C071           0168 EREXIT:
C071 DB30       0169          IN     A,DSTAT        ;DISK STATUS
C073 47         0170          LD     B,A           ;SAVE STATUS
C074 A2         0171          AND    D             ;MASK FOR ERRORS
C075 C9         0172          RET
                                0173 ;
                                0174 ;
C076 3E6F       0175 DSK500: LD     A,6FH         ;TURN ON FAST SEEK
C078 D304       0176          OUT    PARLEL,A      ;
C07A 3E18       0177          LD     A,18H         ;SEEK COMMAND
C07C CD6AC0     0178          CALL    EXECUTE     ;
C07F DB04       0179 DSK540: IN     A,PARLEL       ;FAST SEEK DONE?
C081 E640       0180          AND    40H

```

C083 20FA	0181	JR	NZ,DSK540	
C085 3E7F	0182	LD	A,7FH	;TURN OFF FAST SEEK
C087 D304	0183	OUT	PARLEL,A	
C089 97	0184	SUB	A	;NO ERROR CHECKING, SAY OK
C08A 47	0185	LD	B,A	
C08B C9	0186	RET		
	0187 ;			
	0188 ;			
	0189 ;	READ 1 SECTOR FROM DISK		
	0190 ;			
	0191 ;	INPUT - B CONTAINS DISK (0,1,2,3)		
	0192 ;	E CONTAINS SECTOR		
	0193 ;	A BIT 4 = 1 FOR MAXI		
	0194 ;	HL CONTAINS BUFFER ADDRESS		
	0195 ;			
	0196 ;	OUTPUT - B CONTAINS STATUS		
	0197 ;	Z FLAG IS SET IF NO ERRORS		
	0198 ;	HL PTS PAST BUFFER		
	0199 ;			
	0200 ;	REGISTERS A,F,B,C,D,E,H,L ARE CHANGED		
	0201 ;			
	0202 ;			
C08C CDB6C0	0203 DREAD:	CALL	SETUP	;SET UP FOR READ
C08F C688	0204	ADD	88H	;ADD READ COMMAND TO
C091	0205			;HEAD LOAD FLAG
C091 169C	0206	LD	D,9CH	;ERROR MASK
	0207 ;			
C093 D330	0208	OUT	DCOMMND,A	;OUTPUT READ COMMAND
C095 DB34	0209 DRD250:	IN	A,DFLAGS	;WAIT FOR REQUEST
C097 1F	0210	RRA		;CHECK FOR INTREQ
C098 38D7	0211	JR	C,EREXIT	;END OF SECTOR OR ERROR
C09A EDA2	0212	INI		;READ A BYTE
C09C C295C0	0213	JP	NZ,DRD250	;NOT DONE YET
C09F 18CB	0214	JR	EXCHK	;WAIT FOR INTREQ
	0215 ;			
	0216 ;			
	0217 ;	WRITE A SECTOR TO THE DISK		
	0218 ;			
	0219 ;	INPUT - B CONTAINS DISK (0,1,2,3)		
	0220 ;	E CONTAINS SECTOR		
	0221 ;	A BIT 4 = 1 FOR MAXI		
	0222 ;	HL CONTAINS BUFFER ADDRESS		
	0223 ;			
	0224 ;	OUTPUT - B CONTAINS STATUS		
	0225 ;	Z FLAG IS SET IF NO ERRORS		
	0226 ;	HL PTS PAST BUFFER		
	0227 ;			
	0228 ;	REGISTERS A,F,B,C,D,E,H,L ARE CHANGED		
	0229 ;			
	0230 ;			
C0A1 CDB6C0	0231 DWRITE:	CALL	SETUP	;SET UP FOR WRITE
C0A4 C6A8	0232	ADD	0A8H	;ADD WRITE COMMAND TO
C0A6	0233			;HEAD LOAD FLAG
C0A6 16FC	0234	LD	D,0FCH	;ERROR MASK
C0A8 D330	0235	OUT	DCOMMND,A	;OUTPUT WRITE COMMAND
C0AA DB34	0236 DWR250:	IN	A,DFLAGS	;WAIT FOR REQUEST
C0AC 1F	0237	RRA		;CHECK FOR INTREQ
C0AD 38C2	0238	JR	C,EREXIT	;END OF SECTOR OR ERROR
C0AF EDA3	0239	OUTI		;READ A BYTE
C0B1 C2AAC0	0240	JP	NZ,DWR250	;NOT DONE YET
C0B4 10B6	0241	JR	EXCHK	;WAIT FOR INTREQ
	0242 ;			


```

0243 ;
0244 ;SET UP FOR READ OR WRITE
0245 ;
0246 ;INPUT -- B CONTAINS DISK DRIVE (0,1,2,3)
0247 ;      E CONTAINS SECTOR
0248 ;      A BIT 4 CONTAINS 1 IF MAXI
0249 ;
0250 ;OUTPUT -- D CONTAINS SELECT BYTE
0251 ;      A CONTAINS HEAD LOAD FLAG
0252 ;      B CONTAINS 128 (# OF BYTES)
0253 ;      C CONTAINS DATA PORT
0254 ;
0255 ;REGISTERS A,F,B,C,D ARE CHANGED
0256 ;
0257 ;
C0B6 0258 SETUP:
C0B6 CDCBC0 0259      CALL    SELECT      ;GET SELECT BYTE
C0B9 F680 0260      OR      80H      ;TURN ON AUTO WAIT
C0BB 57 0261      LD      D,A      ;SAVE CONTROL BYTE
C0BC 7B 0262      LD      A,E      ;SECTOR #
C0BD D332 0263      OUT     DSEC,A
0264 ;
0265 ;CHECK WHETHER DISK HEAD LOADED
0266 ;
C0BF DB34 0267      IN      A,DFLAGS  ;READ FLAGS
C0C1 E620 0268      AND     HDLDM    ;HEAD LOADED?
C0C3 7A 0269      LD      A,D      ;CONTROL BYTE
C0C4 D334 0270      OUT     DCONTR,A ;[THIS MUST BE DONE AFTER
0271 ;      ;THE INPUT FROM DFLAGS
0272 ;      ;BECAUSE OF AUTO WAIT]
C0C6 3E04 0273      LD      A,4      ;HEAD NOT LOADED
C0C8 C8 0274      RET      Z
C0C9 97 0275      SUB     A      ;HEAD LOADED
C0CA C9 0276      RET
0277 ;
0278 ;
0279 ;SELECT DISK DRIVE
0280 ;
0281 ;INPUT -- B CONTAINS DISK DRIVE (0,1,2,3)
0282 ;      A BIT 4 CONTAINS 1 IF MAXI
0283 ;
0284 ;OUTPUT -- A CONTAINS SELECT BYTE
0285 ;      B CONTAINS 128
0286 ;      C CONTAINS DATA PORT #
0287 ;
0288 ;REGISTERS A,F,B,C ARE CHANGED
0289 ;
0290 ;
C0CB E610 0291 SELECT: AND     MAXIM      ;GET MAXI FLAG ONLY
C0CD 4F 0292      LD      C,A      ;SAVE FLAG
C0CE 04 0293      INC     B      ;CALCULATE DISK SELECT
C0CF 97 0294      SUB     A
C0D0 37 0295      SCF
C0D1 17 0296 SEL300: RLA
C0D2 10FD 0297      DJNZ    SEL300
C0D4 B1 0298      OR      C      ;MAXI FLAG
C0D5 F620 0299      OR      20H    ;MOTOR ON
C0D7 013380 0300      LD      BC,8000H+DDATA
C0DA C9 0301      RET
0302 ;
0303 ;
0304 ;

```

```

0305 ; CHECK INPUT & RETURN WITH DATA IF READY.
0306 ;
C0DB DB00      0307 CHKIN:  IN      A,STAT
C0DD E340      0308          AND      DAV
C0DF C8        0309          RET      Z
C0E0 DB01      0310          IN      A,DATA
C0E2 C9        0311          RET
0312 ;
0313 ;
0314 ; GET CHARACTER FROM INPUT.
0315 ;
C0E3 CDDBC0    0316 GBYTE:  CALL     CHKIN
C0E6 28FB      0317          JR      Z,GBYTE
C0E8 E67F      0318          AND     7FH
C0EA C9        0319          RET
0320 ;
0321 ;
0322 ; COMMAND
0323 ; CHANGE BAUD RATE OF THE SERIAL PORT
0324 ;
C0EB          0325 INITBR:
C0EB CD29C2    0326          CALL     SKSGCR          ;REQUIRE CR
0327 ;                                     ;CONTINUE BELOW
0328 ;
0329 ;
0330 ; INITIALIZE BAUD RATE OF THE CURRENT DEVICE.
0331 ;
0332 ; PUSH CARRIAGE-RETURN TO SELECT THE PROPER BAUD
0333 ; RATE FOR THE CURRENT TERMINAL. (THE MAXIMUM
0334 ; NUMBER OF CARRIAGE-RETURNS REQUIRED IS FOUR.)
0335 ;
0336 ; ANY OF THE FOLLOWING BAUD RATES CAN BE SELECTED:
0337 ; 19200, 9600, 4800, 2400, 1200, 300, 150, 110.
0338 ;
C0EE 21CAC3    0339 INITBAUD: LD      HL,BAUDRS
C0F1 0E00      0340          LD      C,BAUD
C0F3 3E19      0341          LD      A,19H          ;OCTUPLE THE CLOCK
C0F5 D302      0342 IT1:   OUT      COMMND,A          ;& RESET CURRENT DEVICE
C0F7 EDA3      0343          OUTI
C0F9 CDE3C0    0344          CALL     GBYTE
C0FC CDE3C0    0345          CALL     GBYTE
C0FF FE0D      0346          CP      CR
C101 3E09      0347          LD      A,9          ;SLOW THE CLOCK
C103 20F0      0348          JR      NZ,IT1
C105 C9        0349          RET
0350 ;
0351 ;
0352 ; COMMAND
0353 ; CHANGE LOCATION OF THE SYSTEM STACK
0354 ;
C106          0355 KICKSTK:
C106 CD85C2    0356          CALL     LINCR
C109 1813      0357          JR      LOADIX          ;IX STORES INITIAL SP VALUE
0358 ;
0359 ;
0360 ; -----
0361 ; MONITOR ENTRY POINT
0362 ; -----
0363 ;
0364 ; ENTER MONITOR WITH THE STK FNTR LOADED & WITH
0365 ; DE -> THE DISK FLAGS. (THIS IS ALSO
0366 ; THE TOP OF THE STACK.)

```

```

0367 ;
C10E 0368 MONITR:
C10E CDF7C2 0369 CALL PMSGFOLLOWING
C10E 000D4352 0370 DB CR,CR,'CROMEMCO RDOS','1'+80H
      4F4D454D
      434F2052
      444F53B1

0371 ;
C11E 0372 LOADIX:
C11E 97 0373 SUB A
C11F 12 0374 LD (DE),A ;CLEAR DISK MODE
C120 D5 0375 PUSH DE
C121 DDE1 0376 POP IX ;IX STORES INITIAL SP VALUE
      0377 ;
C123 0378 CLEANSTACK:
C123 DDF9 0379 LD SP,IX ;RE--INITIALIZE SP
      0380 ;
      0381 ;
      0382 ; GET COMMAND.
      0383 ; RETURNS VALUE IN HL & JUMPS TO THAT ADDR.
      0384 ;
C125 CDA9C1 0385 CALL CRLF
C126 2128C1 0386 CMND: LD HL,CMND ;SET-UP RETURN
C12B DDE5 0387 PUSH TX
C12D E3 0388 EX (SP),HL ;RETN ADDR ON STK
C12E 4E 0389 LD C,(HL) ;HL -> DISK FLAGS
C12F CB49 0390 BIT DISKMODE,C
C131 23 0391 INC HL ; -> DISK LETTER
C132 C4ECC2 0392 CALL NZ,PMSG ;DISK MODE PROMPT
C135 CDF7C2 0393 CALL PMSGFOLLOWING
C138 BE 0394 DB ';' + 80H ;THE REGULAR PROMPT
      0395 ;
C139 CDBAC2 0396 CALL SKSG0 ;GET THE COMMAND
C13C 2005 0397 JR NZ,CM6
C13E DD360000 0398 LD (IX),0 ;CR, RESET DISK MODE.
C142 C9 0399 RET
      0400 ;
C143 D641 0401 CM6: SUB 'A'+CASE ; < 'A'?
C145 384E 0402 JR C,ERROR
C147 FE17 0403 CP 'W'-'A'+1 ; > 'W'?
C149 306A 0404 JR NC,ERROR
C14B 5F 0405 LD E,A
C14C 1600 0406 LD D,0
      0407 ;
C14E CDBAC2 0408 CALL SKSG0 ;NEXT COMMAND CHARACTER
C151 FE3B 0409 CP ';'
C153 280D 0410 JR Z,DISKSELECT
C155 EB 0411 EX DE,HL
C156 29 0412 ADD HL,HL ;TIMES 2
C157 11D2C3 0413 LD DE,CMNDTEL
C15A 19 0414 ADD HL,DE ; + TEL ADDR
C15B 5E 0415 LD E,(HL)
C15C 23 0416 INC HL
C15D 56 0417 LD D,(HL)
C15E EB 0418 EX DE,HL
C15F FE4D 0419 CP 'M'+CASE ;(USED IN SUBST & DISPL)
C161 E9 0420 JP (HL)
      0421 ;
      0422 ;
      0423 ; DISK SELECT
      0424 ; ENTER WITH E CONTAINING THE DISK NUMBER
      0425 ;

```

```

C162          0426 DISKSELECT:
C162 7B       0427          LD      A,E           ;DISK NUMBER
C163 FE04     0428          CP      NDRIVES       ;A THROUGH D ONLY
C165 304E     0429          JR      NC,ERROR
C167 43       0430          LD      B,E           ;SAVE DISK #
C168 DDE5     0431          PUSH    IX
C16A E1       0432          POP     HL           ; -> DISK FLAGS
C16B F6B0     0433          OR      C1 SHL DISKMODEI+C1 SHL MAXI+C1 SHL FASTSEEK
C16D 77       0434          LD      (HL),A       ;DISK # & FLAGS
C16E 54       0435          LD      D,H
C16F 5D       0436          LD      E,L
C170 13       0437          INC     DE           ; -> DISK LETTER
C171 78       0438          LD      A,B
C172 C641     0439          ADD     'A'
C174 12       0440          LD      (DE),A       ;DISK LETTER
C175 CD19C2   0441          CALL    GCHR
C178 FE3B     0442          CP      ';'
C17A 2010     0443          JR      NZ,DS2
C17C CBBE     0444          RES     FASTSEEK,(HL) ;NOT FAST SEEK
C17E 13       0445          INC     DE
C17F 12       0446          LD      (DE),A       ;PART OF DISKMODE PROMPT
C180 CD19C2   0447          CALL    GCHR
C183 FE3B     0448          CP      ';'
C185 2005     0449          JR      NZ,DS2
C187 CBA6     0450          RES     MAXI,(HL)     ;MINI FLOPPY
C189 13       0451          INC     DE
C18A 12       0452          LD      (DE),A
C18B 97       0453          SUB     A
                0454 ;
C18C CD29C2   0455 DS2:    CALL    SKSGCR       ;ALSO EXCHG DE & HL
C18F CBFE     0456          SET     7,(HL)       ;MARK END-OF-MSG
                0457 ;
C191 1A       0458          LD      A,(DE)       ;DISK FLAGS
C192 CD3FC0   0459          CALL    DHOME
C195 3E48     0460          LD      A,'H'        ;IN CASE OF HOME ERROR
                0461 ;
C197          0462 DERRCK:
C197 C8       0463          RET     Z           ;IF NO ERROR, DONE
                0464 ;
C198          0465 FERRMSG:
C198 CDF7C2   0466          CALL    PMSGFOLLOWING
C19B 20455252 0467          DB      ' ERR ',' '+80H
                A0
C1A0 CDF1C1   0468          CALL    PCHR        ;ERROR LETTER
C1A3 78       0469          LD      A,B         ;ERROR NUMBER
                0470 ;
                0471 ;
                0472 ; PRINT THE 2 HEX DIGITS IN THE A-REGISTER
                0473 ; AND CLEAN STACK.
                0474 ;
C1A4          0475 P2HXCLEAN:
C1A4 CDD4C2   0476          CALL    P2HEX
C1A7 1810     0477          JR      CLEANV
                0478 ;
                0479 ;
                0480 ; PRINT CRLF
                0481 ;
C1A7          0482 CRLF:
C1A9 3E0D     0483          LD      A,CR
C1AB 1844     0484          JR      PCHR
                0485 ;
                0486 ;

```

```

0487 ; COMMAND
0488 ; EXAMINE INPUT PORT
0489 ;
C1AD          0490 EXMINPUT:
C1AD CD85C2    0491          CALL    LINCR
C1B0 4B        0492          LD      C,E          ;PORT #
C1B1 ED78      0493          IN      A,(C)
C1B3 18EF      0494          JR      P2HXCLEAN    ;PRINT THE VALUE, CRLF
0495 ;
0496 ;
0497 ; ERROR & ESCAPE. RETURNS TO CMND WITH SP
0498 ; RE-INITIALIZED.
0499 ;
C1B5          0500 ERROR:
C1B5 CDF7C2    0501          CALL    FMSGFOLLOWING
C1B8 BF        0502          DB      '?'+80H
C1B9          0503 ESCAPE:
C1B9          0504 CLEANV:
C1B9 C323C1    0505          JP      CLEANSTACK
0506 ;
0507 ;
0508 ; GET NEXT SECTOR FOR THE READ & WRITE DISK
0509 ; ROUTINES. PRESERVES HL AND, BEFORE RETURNING,
0510 ; POPS DE AND BC FROM THE STACK.
0511 ;
C1BC          0512 NEXTSC:
C1BC D9        0513          EXX
C1BD E1        0514          POP     HL          ;RETURN ADDR
C1BE D9        0515          EXX
C1BF D1        0516          POP     DE
C1C0 2805      0517          JR      Z,NS2          ;SKIP IF NO ERROR
C1C2 15        0518          DEC     D          ;TRY AGAIN?
C1C3 28D3      0519          JR      Z,PERRMSG
C1C5 180A      0520          JR      NS4          ;YES. USE OLD MEM PNTR
0521 ;
C1C7 017FFF    0522 NS2:      LD      BC,-81H          ;NO ERROR
C1CA FD09      0523          ADD     IY,BC          ;BUMP THE INCREMENT
C1CC FD23      0524          INC     IY
C1CE E3        0525          EX      (SP),HL          ;USE LATEST MEM PNTR
C1CF 160A      0526          LD      D,10          ;RELOAD RETRIAL COUNTER
0527 ;
C1D1 E1        0528 NS4:      POP     HL          ;MEM PNTR
C1D2 C1        0529          POP     BC
C1D3 79        0530          LD      A,C          ;RELOAD DISK FLAGS
C1D4 D9        0531          EXX
C1D5 E5        0532          PUSH    HL          ;RETURN ADDR
C1D6 D9        0533          EXX
C1D7 C0        0534          RET     NZ          ;IF ERROR, DONE
0535 ;
C1D8 D4A2C3    0536          CALL    NC,PTRKSC          ;IF NEGATIVE, DONE!
C1DB 30DC      0537          JR      NC,CLEANV          ;PRINT TRK, SEC, CLEAN STK.
0538 ;
C1DD 1C        0539          INC     E          ;BUMP SECTOR #
C1DE CDAEC3    0540          CALL    CHKSECNO
C1E1 D0        0541          RET     NC          ;DONE IF # OK
C1E2 DB31      0542          IN      A,DTRACK          ;GET TRACK #
C1E4 3C        0543          INC     A          ;BUMP IT
C1E5 5F        0544          LD      E,A
C1E6 C5        0545          PUSH    BC
C1E7 CD29C3    0546          CALL    SEEKNXT          ;SEEK NEXT TRACK
C1EA C1        0547          POP     BC
C1EB 79        0548          LD      A,C          ;DISK FLAGS

```



```

C1EC 1E01      0549      LD      E,1          ;SECTOR 1
C1EE C9        0550      RET
                0551 ;
                0552 ;
                0553 ; PRINT SPACE.  ALTERS A.
                0554 ;
C1EF 3E20      0555 SPACE: LD      A,' '          ;(CONTINUE BELOW)
                0556 ;
                0557 ;
                0558 ; PRINT THE CHARACTER IN THE A-REGISTER.
                0559 ; (CHKS INPUT FOR ESC.) PRESERVES ALL REGS.
                0560 ;
C1F1 F5        0561 PCHR:  PUSH    AF              ;SAVE THE CHAR
C1F2 E67F      0562 PC1:  AND     7FH
C1F4 FE1B      0563      CP      ESC
C1F6 28C1      0564      JR      Z,ESCAPE
C1F8 FE7D      0565      CP      ALT              ;ALT MODE?
C1FA 28ED      0566      JR      Z,ESCAPE
C1FC CDDBC0    0567      CALL    CHKIN
C1FF 20F1      0568      JR      NZ,PC1
                0569 ;
C201 DB00      0570 PC2:  IN      A,STAT
C203 E680      0571      AND     TBE
C205 28FA      0572      JR      Z,PC2
C207 F1        0573      POP     AF
C208 F5        0574      PUSH    AF
C209 E67F      0575      AND     7FH
C20B D301      0576      OUT     DATA,A
C20D FE0D      0577      CP      CR
C20F 2006      0578      JR      NZ,PC3
C211 CDF7C2    0579      CALL    PMSGFOLLOWING
C214 0A0080    0580      DB      LF,0,80H
C217 F1        0581 PC3:  POP     AF
C218 C9        0582      RET
                0583 ;
                0584 ;
                0585 ; GET CHARACTER. RETURNS IT IN A.
                0586 ; ALTERS F.
                0587 ;
C219 CDE3C0    0588 GCHR:  CALL    GBYTE
C21C CDF1C1    0589      CALL    PCHR
C21F FE61      0590      CP      61H              ;CONVERT LOWER CASE
C221 D9        0591      RET      C              ;TO UPPER.
C222 D620      0592      SUB     20H
C224 C9        0593      RET
                0594 ;
                0595 ;
                0596 ; LOADS HL WITH SOURCE ADDR, BC & DE
                0597 ; WITH THE INCREMENT.  ENDS WITH A CRLF.
                0598 ;
C225 97        0599 L2NCR: SUB     A
                0600 ;
C226 CD64C2    0601 L2NCR: CALL    LD2N
                0602 ;
                0603 ; SKIP INITIAL SPACES.
                0604 ; IF DELIMITER NOT A CR, ERROR
                0605 ;
C229 CDBEC2    0606 SKSGCR: CALL    SKSG              ;WAIT FOR NON-SPACE
C22C 20B7      0607      JR      NZ,ERROR          ;IF NOT CR, ERROR
C22E EB        0608      EX      DE,HL
C22F C9        0609      RET
                0610 ;

```

```

0611 ;
0612 ; PRINT THE NUMBER IN HL, FOLLOWED BY A COLON.
0613 ; PRESERVES ALL REGISTERS EXCEPT A.
0614 ;
C230 CDA9C1 0615 PCADDR: CALL CRLF
0616 ;
C233 CDCFC2 0617 PADDR: CALL FNHL
C236 3E9A 0618 LD A,':'
C238 18B7 0619 JR PCHR
0620 ;
0621 ;
0622 ; COMMAND
0623 ;
C23A CD82C2 0624 VERIF: CALL L3NCR ;GET 3 OPERANDS
0625 ;
0626 ; COMPARES TWO AREAS OF MEMORY. ENTER WITH
0627 ; SOURCE IN HL, DESTINATION IN DE & COUNT
0628 ; IN BC. ALTERS ALL REGISTERS.
0629 ;
C23D 0630 VRFY:
C23D 1A 0631 LD A,(DE)
C23E EDA1 0632 CPI ;COMPARE TO SOURCE
C240 2B 0633 DEC HL
C241 C4CFC2 0634 CALL NZ,PNHL ;PRINT SOURCE ADDR
C244 C4C6C2 0635 CALL NZ,PSNM ; & CONTENTS
C247 EB 0636 EX DE,HL
C248 C4C6C2 0637 CALL NZ,PSNM ; & DEST CONTENTS
C24B C4CCC2 0638 CALL NZ,PSNHL ; & DEST ADDR
C24E C4A9C1 0639 CALL NZ,CRLF
C251 EB 0640 EX DE,HL
C252 23 0641 INC HL
C253 13 0642 INC DE
C254 E0 0643 RET PO ; IF BC=0, DONE.
C255 18E6 0644 JR VRFY
0645 ;
0646 ; COMMAND
0647 ;
C257 0648 MOVE:
C257 CD82C2 0649 CALL L3NCR ;OPERANDS
C25A E5 0650 PUSH HL
C25B D5 0651 PUSH DE
C25C C5 0652 PUSH BC
C25D EDB0 0653 LDIR
C25F C1 0654 POP BC
C260 D1 0655 POP DE
C261 E1 0656 POP HL
C262 18D9 0657 JR VRFY
0658 ;
0659 ;
0660 ;
0661 ; LOAD TWO NUMBERS. LOADS DE WITH THE BEGINNING
0662 ; ADDR, N1. LOADS BC & HL WITH THE INCREMENT
0663 ; NZ-N1+1 (OR WITH NZ IF THE OPR IS 'S').
0664 ; RETURNS WITH LAST DELIMITER IN A.
0665 ;
0666 ;
C264 CD8BC2 0667 LDZN: CALL GNHL ;N1 TO HL, DELIM TO A
C267 EB 0668 EX DE,HL ;SAVE N1 IN DE
C268 CDEBC2 0669 CALL SKSG ;GET NEXT NON-SPACE
C26B FE53 0670 CP 'S'+CASE ;SWATH?
C26D 2005 0671 JR NZ,L2N1
0672 ;

```

```

C26F CD8AC2      0673      CALL    GNHL0      ;YES, INCREMENT TO HL.
C272 1807        0674      JR      L2N2
                  0675 ;
C274 CD8BC2      0676 L2N1:  CALL    GNHL      ;INCREMENT
C277 B7          0677      OR      A          ;CLEAR CY
C278 ED52        0678      SEC      HL,DE      ;N2-N1
C27A 23          0679      INC      HL          ;INCLUDE END POINT
C27B 44          0680 L2N2:  LD      B,H      ;BC GETS THE INCRM
C27C 4D          0681      LD      C,L      ;BC GETS THE INCRM
C27D E5          0682      PUSH    HL
C27E FDE1        0683      POP     IY          ;& SO DOES IY.
C280 C9          0684      RET
                  0685 ;
                  0686 ;
                  0687 ; LOAD 3 OPERANDS. HL GETS THE SOURCE, BC
                  0688 ; THE INCREMENT, AND DE THE 3RD OPERAND.
                  0689 ;
C281 97          0690 L3NCR:  SUB     A
                  0691 ;
C282 CD64C2      0692 L3NCR:  CALL    LD2N
                  0693 ; (CONTINUE BELOW)
                  0694 ;
                  0695 ;
                  0696 ; ENTER WITH SPACE OR THE FIRST DIGIT
                  0697 ; OF A NUMBER IN A. LOADS HL WITH
                  0698 ; WITH A NEW NUMBER & THEN EXCHANGES
                  0699 ; DE & HL. FINISHES WITH A CRLF.
                  0700 ;
C285 CD8EC2      0701 L1NCR:  CALL    GNHL      ;SKIP SPACES, LOAD HL
C288 189F        0702      JR      SKSGCR     ;WAIT FOR A CR
                  0703 ;
                  0704 ;
                  0705 ; CLEARS HL. IF ENTERED WITH HEX CHAR IN A,
                  0706 ; SHIFTS IT INTO HL. O/W, IGNORES LEADING
                  0707 ; SPACES. FIRST CHAR MUST BE HEX. CONTINUES
                  0708 ; SHIFT UNTIL A NON-HEX CHAR RECEIVED & THEN
                  0709 ; RETURNS WITH THE LATTER IN A.
                  0710 ; PRESERVES B,C,D,E.
                  0711 ;
                  0712 ;
C28A 97          0713 GNHL0:  SUB     A
                  0714 ;
C28B C5          0715 GNHL:   PUSH    BC          ;SAVE
C28C 210000      0716      LD      HL,0          ;CLEAR BUFFER
                  0717 ; STRIP LEADING SPACES & GET CHAR
C28F CDBBC2      0718      CALL    SKSG
                  0719 ; FIRST CHAR MUST BE HEX
C292 CDA3C2      0720      CALL    HEXSH      ;IF HEX, SHIFT INTO HL
C295 DAB5C1      0721      JP      C,ERROR     ;O/W, ERROR
C298 CD19C2      0722 GN1:    CALL    GCHR
C29B CDA3C2      0723      CALL    HEXSH      ;IF HEX SHIFT INTO HL
C29E 78          0724      LD      A,B          ;RESTORE CHAR
C29F 30F7        0725      JR      NC,GN1      ;IF HEX, CONTINUE
C2A1 C1          0726      POP     BC          ;IF NON-HEX, DONE
C2A2 C9          0727      RET
                  0728 ;
                  0729 ;
                  0730 ; IF A CONTAINS HEX CHAR, SHIFTS BINARY EQUIVALENT
                  0731 ; INTO HL. IF NOT HEX, RET WITH CY SET. SAVES
                  0732 ; ORIGINAL CHAR IN B
                  0733 ;
C2A3 47          0734 HEXSH:  LD      B,A

```

```

C2A4 D630      0735      SUB      '0'          ; < '0'?
C2A6 D8        0736      RET      C
C2A7 C6E9      0737      ADD      '0'-'C'+CASEJ
C2A9 D8        0738      RET      C
C2AA D6FA      0739      SUB      'A'-'G'
C2AC 3003      0740      JR      NC,HX1          ;OK IF >= 'A'
C2AE C607      0741      ADD      C'A'+CASEJ-C'9'+1J
C2B0 D8        0742      RET      C
C2B1 C60A      0743 HX1:      ADD      '9'+1-'0'
                        0744 ; THE A-REG NOW CONTAINS THE HEX DJGIT IN BINARY.
                        0745 ; (THE HIGH-ORDER NIBBLE OF A IS 0.)
C2B3 29        0746 HXSH4:  ADD      HL,HL          ;SHIFT 4 BITS INTO HL
C2B4 29        0747      ADD      HL,HL
C2B5 29        0748      ADD      HL,HL
C2B6 29        0749      ADD      HL,HL
C2B7 B5        0750      OR      L
C2B8 6F        0751      LD      L,A
C2B9 C9        0752      RET
                        0753 ;
                        0754 ;
                        0755 ; RETURNS WITH A NON-SPACE IN THE A-REG.
                        0756 ; IF ENTERED WITH A-REG CONTAINING A NULL
                        0757 ; OR A SPACE, GETS NEW CHARS UNTIL FIRST
                        0758 ; NON-SPACE OCCURS. ALTERS AF.
                        0759 ;
C2BA 97        0760 SKSG0:  SUB      A
                        0761 ;
C2BB B7        0762 SKSG:   OR      A          ;DOES A CONTAIN NULL?
C2BC CC19C2    0763 SK1:   CALL    Z,GCHR
C2BF FE20      0764      CF      Z0H          ;SPACE?
C2C1 28F9      0765      JR      Z,SK1
C2C3 FE0D      0766      CF      CR
C2C5 C9        0767      RET
                        0768 ;
                        0769 ;
                        0770 ;
                        0771 ; PRINT SPACE FOLLOWED BY THE NUMBER POINTED
                        0772 ; TO BY HL. ALTERS A ONLY.
                        0773 ;
C2C6 CDEFC1    0774 PSNM:   CALL    SPACE
                        0775 ; (CONTINUE BELOW)
                        0776 ;
                        0777 ; PRINTS THE NUMBER POINTED TO BY HL.
                        0778 ; PRESERVES ALL REGISTERS BUT A.
                        0779 ;
C2C9 7E        0780 PNM:    LD      A,(HL)
C2CA 1808      0781      JR      P2HEX
                        0782 ;
                        0783 ;
                        0784 ;
                        0785 ; PRINT THE NUMBER IN HL.
                        0786 ; PRESERVES ALL BUT A.
                        0787 ;
C2CC CDEFC1    0788 PSNHL:  CALL    SPACE
                        0789 ;
C2CF 7C        0790 PNHL:   LD      A,H
C2D0 CDD4C2    0791      (      CALL    P2HEX
C2D3 7D        0792      LD      A,L
                        0793 ;          ;(CONTINUE BELOW)
                        0794 ;
                        0795 ; PRINT THE NUMBER IN THE A-REGISTER.
                        0796 ; PRESERVES ALL REGISTERS.

```

```

C2D4 CDD8C2      0797 ;
C2D7 1F          0798 P2HEX: CALL    P1HEX
C2D8 1F          0799          RRA
C2D9 1F          0800 P1HEX: RRA
C2DA 1F          0801          RRA
C2DB 1F          0802          RRA
C2DC F5          0803          RRA
C2DD E60F        0804          PUSH    AF
C2DE FE0A        0805          AND     0FH          ;MASK
C2DF FE0A        0806          CP      10D          ; <= 9?
C2E1 3802        0807          JR      C,PH1
C2E3 C607        0808          ADD     7            ;A THRU F
C2E5 C630        0809 PH1:    ADD     30H          ;ASCII BIAS
C2E7 CDF1C1      0810          CALL   PCHR          ;PRINT IT
C2EA F1          0811          POP     AF
C2EB C9          0812          RET
0813 ;
0814 ;
0815 ; PRINT MESSAGE, ENTER WITH ADDR OF MSG
0816 ; IN HL.  THE MESSAGE IS TERMINATED
0817 ; AFTER PRINTING A CHARACTER WHOSE
0818 ; PARITY BIT WAS SET.
0819 ; PRESERVES FLAGS, INCREMENTS HL.
0820 ;
0821 ;
0822 ;
C2EC F5          0823 PMSG:   PUSH    AF            ;SAVE
C2ED 7E          0824 PS1:    LD      A,(HL)
C2EE 23          0825          INC     HL
C2EF CDF1C1      0826          CALL   PCHR
C2F2 17          0827          RLA              ;LAST CHARACTER?
C2F3 30F8        0828          JR      NC,PS1      ;IF NOT, LOOP
C2F5 F1          0829          POP     AF
C2F6 C9          0830          RET
0831 ;
0832 ;
0833 ; PRINTS THE MESSAGE FOLLOWING THE CALL
0834 ; TO THIS ROUTINE.
0835 ; PRESERVES ALL REGISTERS
0836 ;
C2F7            0837 PMSGFOLLOWING:
C2F7 E3          0838          EX      (SP),HL
C2F8 CDECC2      0839          CALL   PMSG
C2FB E3          0840          EX      (SP),HL
C2FC C9          0841          RET
0842 ;
0843 ;
0844 ; COMMAND
0845 ;
0846 ; GO <ADDR>
0847 ; EXECUTION BEGINS AT ADDR.
0848 ;
C2FD            0849 GO:
C2FD E1          0850          POP     HL            ;CLEAN STACK
C2FE CD85C2      0851          CALL   L1NCR          ;GET ADDR
C301 EB          0852          EX      DE,HL
C302 E9          0853          JP      (HL)
0854 ;
0855 ;
0856 ; COMMAND.  DISPLAY MEMORY.
0857 ;
0858 ; DN <STARTING ADDR> <ENDING ADDR OR SWATH>

```



```

0359 ;
C303 0860 DSPM1:
C303 2036 0861 JR NZ,ERRORV ;IF NOT 'M', ERROR
C305 CD25C2 0862 CALL L2NCR0 ;GET OPERANDS
C308 1610 0863 DSPM1: LD D,16 ;BYTE COUNT
C30A CD30C2 0864 CALL PCADDR ;ADDRESS
C30D CDC6C2 0865 DM2: CALL PSNM ;MEM CONTENTS
C310 EDA1 0866 CPI ;INC HL & DEC BC
C312 E2A9C1 0867 JF PO,CRLF
C315 15 0868 DEC D
C316 28F0 0869 JR Z,DSPM1
C318 7A 0870 LD A,D
C319 E403 0871 AND 3
C31E CCEFC1 0872 CALL Z,SPACE
C31E 18ED 0873 JR DM2
0874 ;
0875 ;
C320 0876 SHANDLER:
C320 281C 0877 JR Z,SUBSM ;IF 'M', SUBSM
0878 ;
0879 ;
0880 ; DISK SEEK
0881 ;
C322 0882 SEEKR:
C322 CB69 0883 BIT DISKMODE,C
C324 2815 0884 JR Z,ERRORV
C326 CD85C2 0885 CALL L1NCR ;E = TRACK #
C329 0886 SEEKNXT:
C329 3E4C 0887 LD A,76 ;MAX TRACK #, MAXI DISK
C32B 1627 0888 LD D,39 ;MAX TRACK #, MINI DISK
C32D CDB2C3 0889 CALL CHKNO ;CHECK #
C330 3809 0890 JR C,ERRORV
C332 53 0891 LD D,E ;TRACK #
C333 CD54C0 0892 CALL DSEEK
C336 3E53 0893 LD A,'S' ;IN CASE OF SEEK ERROR
0894 ;
C338 C397C1 0895 DERRCKV: JF DERRCK ;DISK ERROR CHECK
0896 ;
0897 ;
C33B C3B5C1 0898 ERRORV: JF ERROR
0899 ;
0900 ;
0901 ; COMMAND. SUBSTITUTE MEMORY LOCATION.
0902 ;
0903 ; SM <ADDR>
0904 ;
C33E 0905 SUBSM:
C33E 97 0906 SUB A
C33F CD85C2 0907 CALL L1NCR
C342 EB 0908 EX DE,HL ;HL GETS ADDR
C343 CD30C2 0909 SM1: CALL Z,PCADDR
C346 CCEFC1 0910 CALL Z,SPACE
0911 ; PRINT CURRENT VALUE, REQUEST NEW VALUE &
0912 ; PRINT IT IF GIVEN
C349 CDC9C2 0913 CALL PNM ;PRINT (HL)
C34C CDF7C2 0914 CALL PMSGFOLLOWING
C34F AE 0915 DB ','+80H ;THE PROMPT
C350 CD19C2 0916 CALL GCHR
C353 FE2F 0917 CP ','+1 ;IF <= ',',
C355 DCF1C1 0918 CALL C,PCHR ;NO SUBSTITUTION.
C358 3806 0919 JR C,SM2
C35A EB 0920 EX DE,HL

```

```

C35B CD8EC2      0921      CALL      GNHL      ;GET NEW VALUE
C35E EB          0922      EX        DE,HL
C35F 73          0923      LD        (HL),E
C360 FE0D        0924 SM2:    CP        CR
C362 C4EFC1      0925      CALL      NZ,SPACE
                  0926 ;
C365 C8          0927      RET        Z          ;IF CR, DONE.
C366 23          0928      INC        HL
C367 3E07        0929      LD        A,7          ;PRINT ADDRESS IF IT
C369 A5          0930      AND        L          ;IS A MULTIPLE OF 8
C36A 18D7        0931      JR        SM1
                  0932 ;
                  0933 ;
C36C             0934 RHANDLER:
C36C FE44        0935      CP        'D'+CASE
C36E 20CB        0936      JR        NZ,ERRORV
                  0937 ;
                  0938 ; READ DISK
                  0939 ;
C370             0940 READDR:
C370 CD94C3      0941      CALL      SECSETUP
C373 C5          0942 RD2:    PUSH     BC
C374 E5          0943      PUSH     HL
C375 D5          0944      PUSH     DE
C376 CD8CC0      0945      CALL      DREAD
C379 3E52        0946      LD        A,'R'          ;IN CASE OF READ ERROR
C37B CDECC1      0947      CALL      NEXTSC          ;NEXT SECTOR [POPS STK.]
C37E 18F3        0948      JR        RD2
                  0949 ;
                  0950 ;
C380             0951 WHANDLER:
C380 FE44        0952      CP        'D'+CASE
C382 20E7        0953      JR        NZ,ERRORV
                  0954 ;
                  0955 ; WRITE DISK
                  0956 ;
C384             0957 WRITDR:
C384 CD94C3      0958      CALL      SECSETUP
C387 C5          0959 WD2:    PUSH     BC
C388 E5          0960      PUSH     HL
C389 D5          0961      PUSH     DE
C38A CDA1C0      0962      CALL      DWRITE
C38D 3E57        0963      LD        A,'W'          ;IN CASE OF WRITE ERROR
C38F CDECC1      0964      CALL      NEXTSC          ;[POPS STACK]
C392 18F3        0965      JR        WD2
                  0966 ;
                  0967 ;
                  0968 ; GET MEMORY ADDRESS, SECTOR # AND CHECK IT,
                  0969 ; AND LOAD B & C.
                  0970 ;
C394             0971 SECSETUP:
C394 CB69        0972      BIT        DISKMODE,C
C396 28A3        0973      JR        Z,ERRORV
C398 C5          0974      PUSH     BC
C399 CD81C2      0975      CALL      L3NCR0          ;BUFFER ADDR & SEC #
C39C C1          0976      POP      BC
C39D CDAEC3      0977      CALL      CHKSECNO
C3A0 3899        0978      JR        C,ERRORV
                  0979 ;
                  0980 ;
                  0981 ; PRINT TRACK & SECTOR #'S
                  0982 ;

```

```

C3A2          0983 PTRKSC:
C3A2 DE31     0984          IN      A,DTRACK
C3A4 57       0985          LD      D,A
C3A5 EB       0986          EX      DE,HL
C3A6 CDCCC2   0987          CALL   PSNHL          ;PRINT TRK & SEC
C3A9 EB       0988          EX      DE,HL
C3AA 79       0989          LD      A,C          ;DISK FLAGS
C3AB 160A     0990          LD      D,10         ;# OF RETRIALS
C3AD C9       0991          RET
              0992 ;
              0993 ;
C3AE          0994 CHKSECNO:
C3AE 3E1A     0995          LD      A,26         ;MAX SEC #, MAXI DISK
C3B0 1612     0996          LD      D,18         ;MAX SEC #, MINI DISK
              0997 ;
              0998 ;
C3B2          0999 CHKNO:
C3B2 CB61     1000          BIT     MAXI,C
C3B4 2001     1001          JR      NZ,CN2
C3B6 7A       1002          LD      A,D
C3B7 BB       1003 CN2:    CP      E
C3B8 D8       1004          RET      C
C3B9 79       1005          LD      A,C
C3BA E603     1006          AND     NDRIVES-1
C3BC 47       1007          LD      B,A          ;DISK #
C3BD 79       1008          LD      A,C          ;DISK FLAGS
C3BE C9       1009          RET
              1010 ;
              1011 ;
              1012 ; COMMAND
              1013 ; OUT <DATA-BYTE> <PORT NNUMBER>
              1014 ;
C3BF CD8BC2   1015 DUTF:   CALL     GNHL
C3C2 EB       1016          EX      DE,HL          ;E GETS DATA
C3C3 CD85C2   1017          CALL   LINCX          ;GET PORT NUMBER
              1018 ;
C3C6 4B       1019          LD      C,E          ; TO C
C3C7 ED69     1020          OUT     (C),L
C3C9 C9       1021          RET
              1022 ;
              1023 ;
              1024 ; BAUD RATES:
              1025 ; 19200, 9600, 4800, 2400, 1200, 300, 150, 110.
              1026 ;
              1027 ;
C3CA 90C0A090 1028 BAUDRS: DE      90H,0C0H,0A0H,90H,88H,84H,82H,1
      88848201
              1029 ;
              1030 ;
C3D2          1031 CMNDTBL:
C3D2 B5C1     1032          DW     ERROR          ;A
C3D4 15C0     1033          DW     BOOTMC         ;BOOT CDOS
C3D6 B5C1     1034          DW     ERROR          ;C
C3D8 03C3     1035          DW     DSPM          ;DISPLAY MEMORY
C3DA ADC1     1036          DW     EXMINPUT       ;EXAMINE INPUT PORT
C3DC B5C1     1037          DW     ERROR          ;F
C3DE FDC2     1038          DW     GO            ;GO [TRANSFER OF CONTROL]
C3E0 B5C1     1039          DW     ERROR          ;H
C3E2 EBC0     1040          DW     INITER        ;INITIALIZE BAUD RATE
C3E4 B5C1     1041          DW     ERROR          ;J
C3E6 06C1     1042          DW     KICKSTK        ;KICK SYSTEM STACK
C3E8 B5C1     1043          DW     ERROR          ;L

```

```

C3EA 57C2      1044      DW      MOVE      ;MOVE A BLOCK OF MEMORY
C3EC B5C1      1045      DW      ERROR      ;N
C3EE BFC3      1046      DW      OUTF      ;OUTPUT
C3F0 B5C1      1047      DW      ERROR      ;P
C3F2 B5C1      1048      DW      ERROR      ;Q
C3F4 6CC3      1049      DW      RHANDLER    ;READ DISK
C3F6 20C3      1050      DW      SHANDLER    ;SUBSTITUTE MEM; SEEK TRACK
C3F8 B5C1      1051      DW      ERROR      ;T
C3FA B5C1      1052      DW      ERROR      ;U
C3FC 3AC2      1053      DW      VERIF      ;VERIFY BLOCKS OF MEMORY
C3FE 80C3      1054      DW      WHANDLER    ;WRITE DISK
                1055 ;
                1056 ;
(C3FF)        1057 LASTBYTE: EQU    $-1
                1058 ;

```

0000 ERRORS

CROMEMCO CDOS Z80 ASSEMBLER V.1.4A

SYMBOL TABLE

ALT	007D	BAUD	0000	BAUDRS	C3CA	BOOTDK	C018
BOOTMC	C015	BOOTSW	0090	BOT200	C01C	BOT300	C024
BOT500	C03A	CASE	0000	CHKIN	C0DB	CHKNO	C3B2
CHKSEC	C3AE	CLEANS	C123	CLEANV	C1B9	CM6	C143
CMND	C128	CMNDTB	C3D2	CN2	C3B7	COMMND	0002
CR	000D	CRLF	C1A9	DATA	0001	DAV	0040
DCOMMN	0030	DCONTR	0034	DDATA	0033	DERCKV	C338
DERRCK	C197	DFLAGS	0034	DHOME	C03F	DISKMO	0005
DISKSE	C162	DM2	C30D	DRD250	C095	DREAD	C08C
DS2	C18C	DSEC	0032	DSEEK	C054	DSK500	C076
DSK540	C07F	DSPM	C303	DSPM1	C308	DSTAT	0030
DTRACK	0031	DWR250	C0AA	DWRITE	C0A1	EREXTT	C071
ERROR	C1B5	ERRORV	C33B	ESC	001B	ESCAPE	C1B9
EXCCHK	C06C	EXECUT	C06A	EXMINF	C1AD	FASTSE	0007
GBYTE	C0E3	GCHR	C219	GN1	C298	GNHL	C28B
GNHL0	C28A	GO	C2FD	HOLDM	0020	HEXSH	C2A3
HX1	C2B1	HXSH4	C2B3	IMASK	0003	INITBA	C0EE
INITBR	C0EE	IT1	C0F5	KICKST	C106	L1NCR	C285
L2N1	C274	L2N2	C27B	L2NCR	C226	L2NCR0	C225
L3NCR	C282	L3NCR0	C281	LASTBY	C3FF	LD2N	C264
LF	000A	LOADIX	C11E	MAXI	0004	MAXIM	0010
MONITR	C10E	MOVE	C257	NDRIVE	0004	NEXTSC	C1BC
NS2	C1C7	NS4	C1D1	OUTP	C3BF	P1HEX	C2D8
P2HEX	C2D4	P2HXCL	C1A4	PADDR	C233	PARLEL	0004
PC1	C1F2	PC2	C201	PC3	C217	PCADDR	C230
PCHR	C1F1	PERRMS	C198	PH1	C2E5	PMSG	C2EC
PMSGFO	C2F7	PNHL	C2CF	PNM	C2C9	PS1	C2ED
PSNHL	C2CC	PSNM	C2C6	PTRKSC	C3A2	RD2	C373
READDR	C370	RHANDL	C36C	SECSET	C394	SEEKNX	C329
SEEKR	C322	SEL300	C0D1	SELECT	C0CB	SETUP	C0B6
SHANDL	C320	SK1	C2BC	SKSG	C2BB	SKSG0	C2DA
SKSGCR	C229	SM1	C343	SM2	C360	SPACE	C1EF
STACK	007C	START	C000	STAT	0000	SUBSM	C33E
TBF	0080	VERIF	C23A	VRFY	C23D	WD2	C387
WHANDL	C380	WRITDR	C384				

CROMEMCO CROSS REFERENCE LISTING V.1.0 FOR FILE RDOS

ALT	0055	0565							
BAUD	0037	0340							
BAUDRS	1028	0339							
BOOTDK	0082	0069							
BOOTMC	0076	1033							
BOOTSW	0039	0068							
BOT200	0085	0087							
BOT300	0090	0105							
BOT500	0103	0076							
CASE	0050	0401	0419	0670	0737	0741	0935	0952	
CHKIN	0307	0316	0567						
CHKNO	0999	0889							
CHKSEC	0994	0540	0277						
CLEANS	0378	0505							
CLEANV	0504	0477	0537						
CM4	0401	0397							
CMND	0386	0386							
CMNDTB	1031	0413							
CN2	1003	1001							
COMMND	0046	0342							
CR	0052	0346	0370	0370	0483	0577	0766	0924	
CRLF	0482	0385	0615	0639	0867				
DATA	0045	0310	0576						
DAV	0047	0303							
DCOMMN	0029	0084	0162	0208	0235				
DCONTR	0033	0120	0149	0270					
DDATA	0031	0300							
DERCKV	0895								
DERRCK	0462	0895							
DFLAGS	0032	0067	0165	0209	0236	0267			
DHOME	0119	0095	0459						
DISKMO	0023	0390	0433	0883	0972				
DISKSE	0426	0410							
DM2	0865	0873							
DRD250	0209	0213							
DREAD	0203	0101	0945						
DS2	0455	0443	0449						
DSEC	0030	0263							
DSEEK	0147	0892							
DSK500	0175	0154							
DSK540	0179	0181							
DSPM	0860	1035							
DSPM1	0863	0869							
DSTAT	0028	0085	0169						
DTRACK	0034	0542	0984						
DWR250	0236	0240							
DWRITE	0231	0962							
EREXIT	0168	0211	0238						
ERROR	0500	0402	0404	0429	0607	0721	0898	1032	1034
		1043	1045	1047	1048	1051	1052	1037	1039
								1041	
ERRORV	0898	0861	0884	0890	0936	0953	0973	0978	
ESC	0054	0563							
ESCAPE	0503	0564	0566						
EXCCHK	0164	0167	0214	0241					
EXECUT	0161	0126	0128	0157	0178				
EXMINP	0490	1036							
FASTSE	0022	0433	0444						

GEYTE	0316	0317	0344	0345	0588
GCHR	0588	0441	0447	0722	0763 0916
GN1	0722	0725			
GNHL	0715	0667	0676	0701	0921 1015
GNHL0	0713	0673			
GO	0849	1038			
HDLDM	0041	0268			
HEXSH	0734	0720	0723		
HX1	0743	0740			
HXSH4	0746				
IMASK	0036	0066			
INITEA	0339	0064			
INITER	0325	1040			
IT1	0342	0348			
KICKST	0355	1042			
L1NCR	0701	0356	0491	0851	0885 0907 1017
L2N1	0676	0671			
L2N2	0680	0674			
L2NCR	0601				
L2NCR0	0599	0862			
L3NCR	0622	0624	0649		
L3NCR0	0690	0975			
LASTBY	1057				
LD2N	0667	0601	0692		
LF	0053	0580			
LOADIX	0372	0357			
MAXI	0024	0039	0104	0433	0450 1000
MAXIM	0040	0122	0155	0291	
MONITR	0368	0070			
MOVE	0648	1044			
NDRIVE	0018	0428	1006		
NEXTSC	0512	0947	0964		
NS2	0522	0517			
NS4	0528	0520			
OUTP	1015	1046			
P1HEX	0800	0798			
P2HEX	0798	0476	0781	0791	
P2HXCL	0475	0494			
PADDR	0617				
PARLEL	0038	0124	0176	0179	0183
PC1	0562	0568			
PC2	0570	0572			
PC3	0581	0578			
PCADDR	0615	0864	0909		
PCHR	0561	0468	0484	0589	0619 0810 0826 0918
PERRMS	0465	0519			
PH1	0809	0807			
PMSG	0823	0392	0837		
PMSGFO	0837	0369	0393	0466	0501 0579 0914
PNHL	0790	0617	0634		
PNM	0780	0913			
PS1	0824	0828			
PSNHL	0788	0638	0987		
PSNM	0774	0635	0637	0865	
PTRKSC	0983	0536			
RD2	0942	0948			
READDR	0940				
RHANDL	0934	1049			
SECSET	0971	0941	0958		
SEEKNX	0886	0546			
SEEKR	0887				
SEL300	0226	0297			

SELECT	0291	0119	0148	0259
SETUP	0258	0203	0231	
SHANDL	0876	1050		
SK1	0763	0765		
SKSG	0762	0606	0669	0718
SKSG0	0760	0396	0408	
SKSGCR	0606	0077	0326	0455 0702
SM1	0909	0931		
SM2	0924	0919		
SPACE	0555	0774	0788	0872 0910 0925
STACK	0008	0060		
START	0060			
STAT	0044	0307	0570	
SUBSM	0905	0877		
TBE	0048	0571		
VERIF	0624	1053		
VRFY	0630	0644	0657	
WD2	0959	0965		
WHANDL	0951	1054		
WRITDR	0957			